NYC Traffic Delay Time Prediction

05th January 2023

Ahmed Dider Rahat

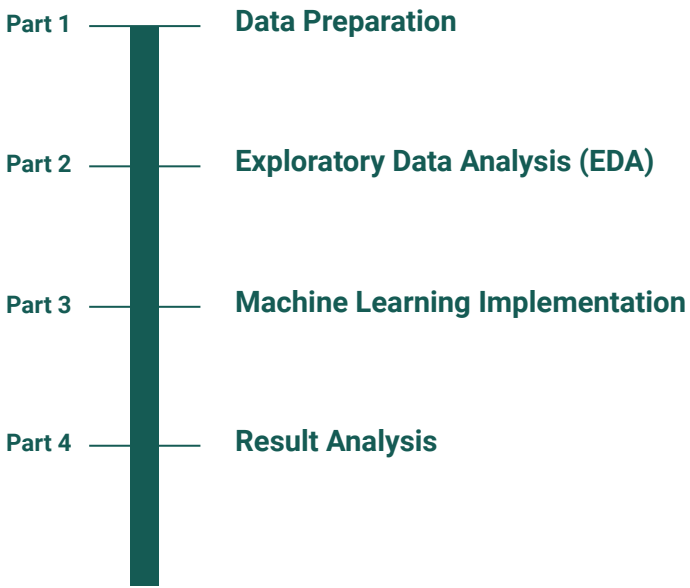
Goal of the Project:

1. Analyze the NYC bus dataset.
2. Implement a ML model to predict the delay time.

Data set:

1. The dataset is collected from a kaggle.
2. The data set is huge and around 26 M data points.
3. There are 17 variables in the data set.
4. No such column for delay time.

Project Flow:



Data Preparation:

1. Load the data set with pandas dataframe.

	RecordedAtTime	DirectionRef	PublishedLineName	OriginName	OriginLat	OriginLong	DestinationName	DestinationLat	DestinationLong
0	2017-06-01 00:03:34	0.0	B8	4 AV/95 ST	40.616104	-74.031143	BROWNSVILLE ROCKAWAY AV	40.656048	-73.907379
1	2017-06-01 00:03:43	1.0	S61	ST GEORGE FERRY/S61 & S91	40.643169	-74.073494	S I MALL YUKON AV	40.575935	-74.167686
2	2017-06-01 00:03:49	0.0	Bx10	E 206 ST/BAINBRIDGE AV	40.875008	-73.880142	RIVERDALE 263 ST	40.912376	-73.902534
3	2017-06-01 00:03:31	0.0	Q5	TEARDROP/LAYOVER	40.701748	-73.802399	ROSEDALE LIRR STA via MERRICK	40.666012	-73.735939
4	2017-06-01 00:03:22	1.0	Bx1	RIVERDALE AV/W 231 ST	40.881187	-73.909340	MOTT HAVEN 136 ST via CONCOURSE	40.809654	-73.928360

Data Preparation (Cont.):

2. The Columns/Variable of the dataset:

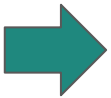
```
# get columns name
list(df.columns)
```

```
['RecordedAtTime',
 'DirectionRef',
 'PublishedLineName',
 'OriginName',
 'OriginLat',
 'OriginLong',
 'DestinationName',
 'DestinationLat',
 'DestinationLong',
 'VehicleRef',
 'VehicleLocation.Latitude',
 'VehicleLocation.Longitude',
 'NextStopPointName',
 'ArrivalProximityText',
 'DistanceFromStop',
 'ExpectedArrivalTime',
 'ScheduledArrivalTime']
```

Data Preparation (Cont.):

3. Rename the Column.

```
['RecordedAtTime',
 'DirectionRef',
 'PublishedLineName',
 'OriginName',
 'OriginLat',
 'OriginLong',
 'DestinationName',
 'DestinationLat',
 'DestinationLong',
 'VehicleRef',
 'VehicleLocation.Latitude',
 'VehicleLocation.Longitude',
 'NextStopPointName',
 'ArrivalProximityText',
 'DistanceFromStop',
 'ExpectedArrivalTime',
 'ScheduledArrivalTime']
```



```
['recorded_at',
 'direction',
 'line_name',
 'org_name',
 'org_lat',
 'org_long',
 'dest_name',
 'dest_lat',
 'dest_long',
 'vech_name',
 'vech_lat',
 'vech_long',
 'next_point_name',
 'arrivial_app',
 'dist_from_stop',
 'expected_arr_time',
 'schedule_arr_time']
```

Data Preparation (Cont.):

4. Precondition for calculating **Delay Time**:
    - Delay time is the subtraction of schedule_arr_time and expected_arr_time.
    - Count the null values in these columns.
    - There are approximately **4.5 M** null values in these variable.

```
Null Counts:
False    22058456
True      4462260
dtype: int64
```

    - So, drop those rows.

Data Preparation (Cont.):

5. Removing duplicate rows if exist.
6. Take a sample of 1 M rows for further operations.

Data Preparation (Cont.):

5. Removing duplicate rows if exist.
6. Take a sample of 1 M rows for further operations.
7. Expected time and schedule time are:

	expected_arr_time	schedule_arr_time
3497201	2017-10-17 01:06:19	25:10:29
505529	2017-06-03 01:57:44	25:44:34
4491653	2017-10-21 01:35:28	25:28:19
4937382	2017-08-23 01:51:35	25:06:28
3493471	2017-12-17 01:02:27	25:12:13

In Schedule Time:

- date is missing.
- Time starts with 24, 25, 26 [For, next day 00, 01, 02 AM]

Data Preparation (Cont.):

8. Calculating Delay time:

```
Expected : 2017-08-22 16:31:00 | schedule : 16:30:22
Expected : 2017-08-21 09:38:29 | schedule : 09:39:33
Expected : 2017-12-18 00:34:10 | schedule : 24:27:26
Expected : 2017-08-18 01:52:26 | schedule : 25:52:54
Expected : 2017-06-27 23:58:46 | schedule : 24:00:44
Expected : 2017-12-02 23:49:42 | schedule : 24:00:50
Expected : 2017-06-18 23:34:11 | schedule : 01:23:21
Expected : 2017-06-18 23:43:50 | schedule : 00:57:48
```

Expected	Schedule	Schedule DateTime
2017-08-22 16:31:00	16:30:22	2017-08-22 16:30:22
2017-12-18 00:34:10	24:27:26	2017-12-18 00:27:26
2017-06-27 23:58:46	24:00:44	2017-06-28 00:00:44
2017-06-18 23:34:11	01:23:21	2017-06-19 01:23:21

Data Preparation (Cont.):

9. Calculate the weekend status (**True:** Week End, **False:** Week Day).

10. Calculate day of year, month number, day of the month, and time of the day (Min).

11. Calculate delay time = [schedule_arr_time – expected_arr_time].

Data Preparation (Cont.):

12. Final data look:

	recorded_at	direction	line_name	org_name	org_lat	org_long	dest_name	dest_lat	dest_long	vech_name	...	arrivial_app
4845711	2017-08-22 16:30:42	1.0	B65	ST JOHNS PL/RALPH AV	40.670227	-73.923233	DNTWN BKLYN FULTON MALL	40.690514	-73.987724	NYCT_6815	...	approaching
4512798	2017-08-21 09:38:07	0.0	B3	25 AV/HARWAY AV	40.593021	-73.992180	BERGEN BCH E 71 ST via AV U	40.619881	-73.907265	NYCT_7166	...	at stop
5690526	2017-06-26 19:05:22	1.0	X10	E 57 ST/3 AV	40.760429	-73.967674	PT RICHMOND via NARROWS RD via GANNON AV	40.633698	-74.129776	NYCT_2648	...	approaching
4078859	2017-08-18 19:02:47	0.0	M15-SBS	SOUTH FERRY/TERMINAL	40.702171	-74.013535	SELECT BUS SERVICE 125ST via 1 AV	40.803150	-73.932266	NYCT_1255	...	< 1 stop away
6498099	2017-06-30 06:02:29	1.0	Bx32	VA HOSPITAL/VA HOSPITAL	40.867352	-73.905624	MOTT HAVEN 136 ST	40.809654	-73.928360	NYCT_7764	...	approaching

Data Preparation (Cont.):

12. Final data look (Cont.):

arrivial_app	dist_from_stop	expected_arr_time	schedule_arr_time	weekend_status	day_of_year	month_number	day_of_month	time_of_day	delays
approaching	65.0	2017-08-22 16:31:00	2017-08-22 16:30:22	False	234	8	22	990	0.63
at stop	14.0	2017-08-21 09:38:29	2017-08-21 09:39:33	False	233	8	21	579	0.00
approaching	95.0	2017-06-26 19:06:01	2017-06-26 19:07:08	False	177	6	26	1147	0.00
< 1 stop away	666.0	2017-08-18 19:07:08	2017-08-18 19:01:00	False	230	8	18	1141	6.13
approaching	94.0	2017-06-30 06:03:00	2017-06-30 06:04:20	False	181	6	30	364	0.00

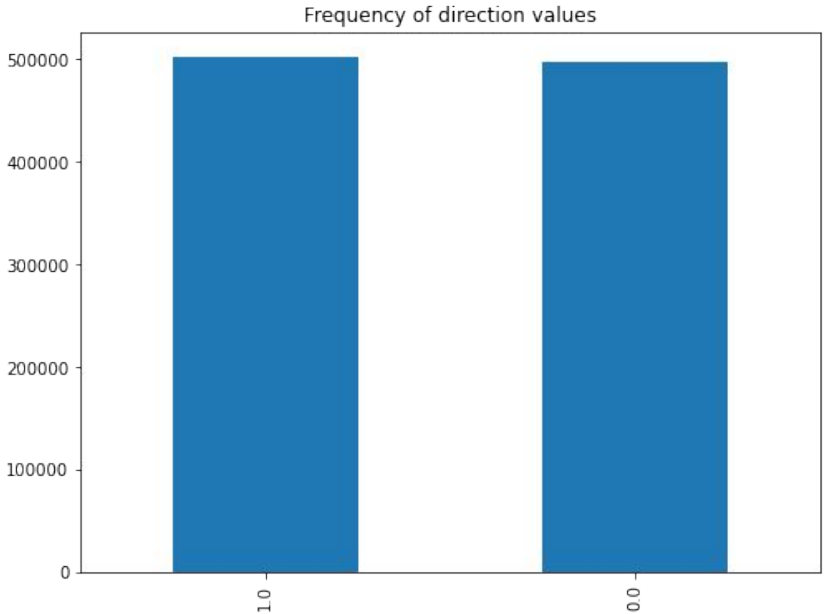
Exploratory Data Analysis (EDA):

1. After data preparation, the size of the columns become: **(1000000, 23)**
2. Check Null values in each columns:

```
In row recorded_at, total number of Null values: 0
In row direction, total number of Null values: 0
In row line_name, total number of Null values: 0
In row org_name, total number of Null values: 0
In row org_lat, total number of Null values: 0
In row org_long, total number of Null values: 0
In row dest_name, total number of Null values: 0
In row dest_lat, total number of Null values: 0
In row dest_long, total number of Null values: 0
In row vech_name, total number of Null values: 0
In row vech_lat, total number of Null values: 0
In row vech_long, total number of Null values: 0
In row next_point_name, total number of Null values: 0
In row arrivial_app, total number of Null values: 0
In row dist_from_stop, total number of Null values: 0
In row expected_arr_time, total number of Null values: 0
In row schedule_arr_time, total number of Null values: 0
In row weekend_status, total number of Null values: 0
In row day_of_year, total number of Null values: 0
In row month_number, total number of Null values: 0
In row day_of_month, total number of Null values: 0
In row time_of_day, total number of Null values: 0
In row delays, total number of Null values: 0
```

Exploratory Data Analysis (Cont.):

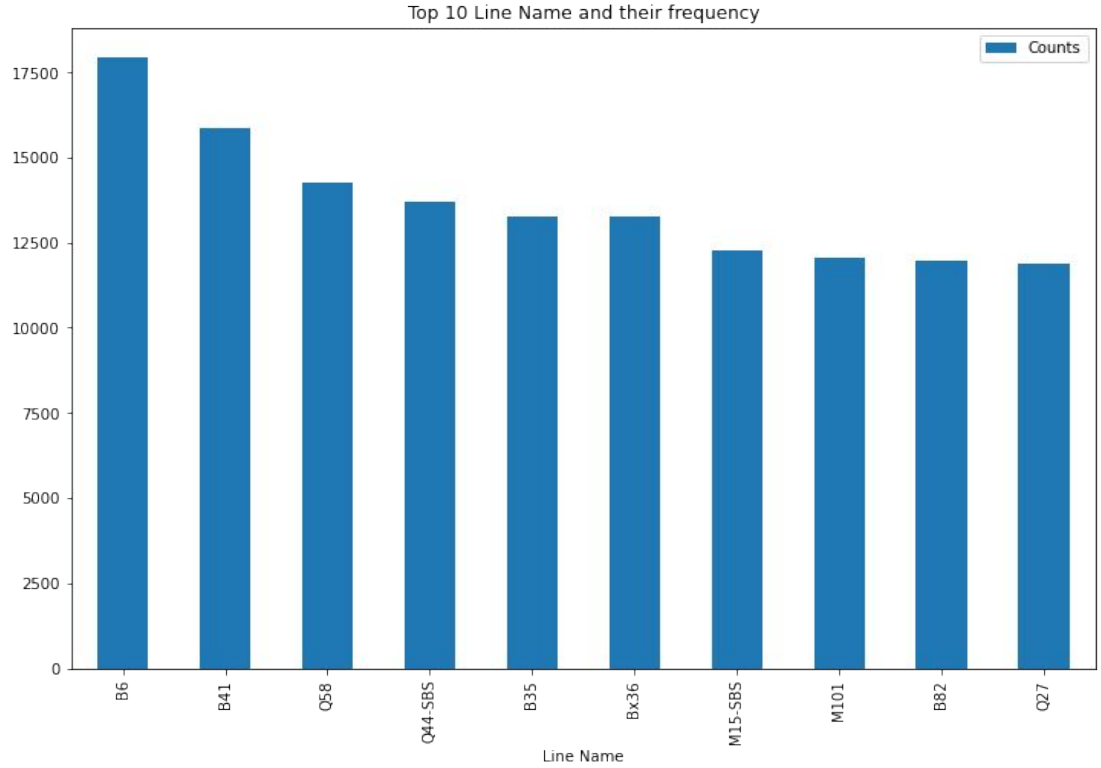
3. Analysis variable direction:



Exploratory Data Analysis (Cont.):

4. **Analysis variable ‘Line Name’:** Total number of Unique line name is **242.**
5. **Top 10 frequent line name:**

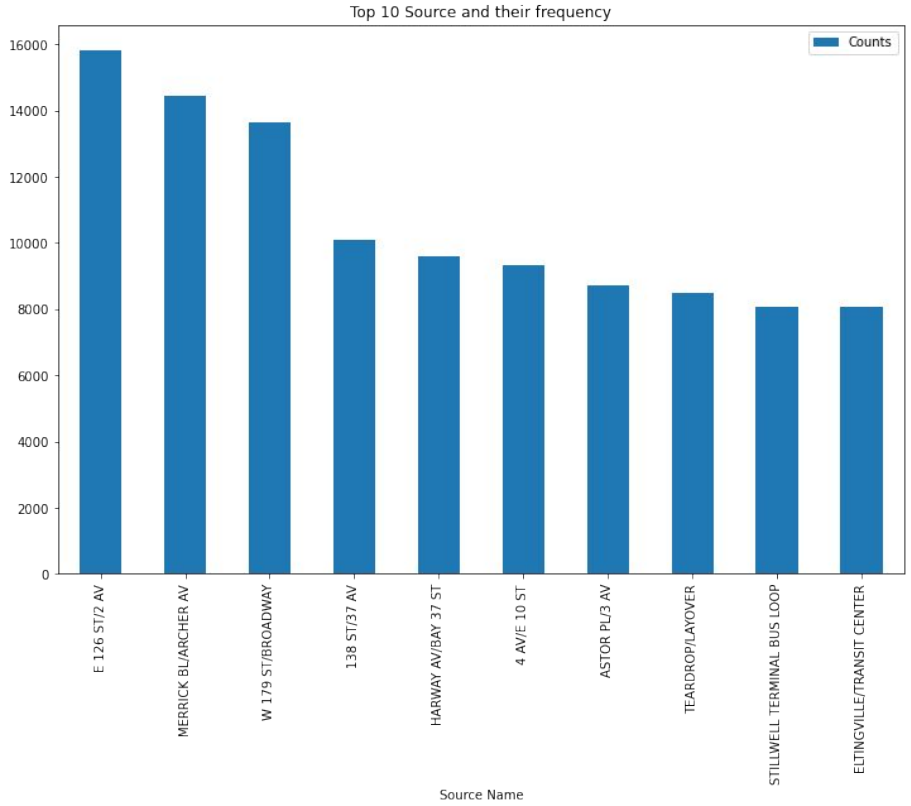
	Line Name	Counts
0	B6	17915
1	B41	15836
2	Q58	14247
3	Q44-SBS	13679
4	B35	13271
5	Bx36	13260
6	M15-SBS	12282
7	M101	12031
8	B82	11960
9	Q27	11877



Exploratory Data Analysis (Cont.):

6. **Analysis variable 'Source Stop':** Total number of Unique **source stop** is **628.**
7. **Top 10 frequent source stop name:**

	Source Name	Counts
0	E 126 ST/2 AV	15808
1	MERRICK BL/ARCHER AV	14427
2	W 179 ST/BROADWAY	13639
3	138 ST/37 AV	10101
4	HARWAY AV/BAY 37 ST	9614
5	4 AV/E 10 ST	9328
6	ASTOR PL/3 AV	8704
7	TEARDROP/LAYOVER	8494
8	STILLWELL TERMINAL BUS LOOP	8081
9	ELTINGVILLE/TRANSIT CENTER	8071

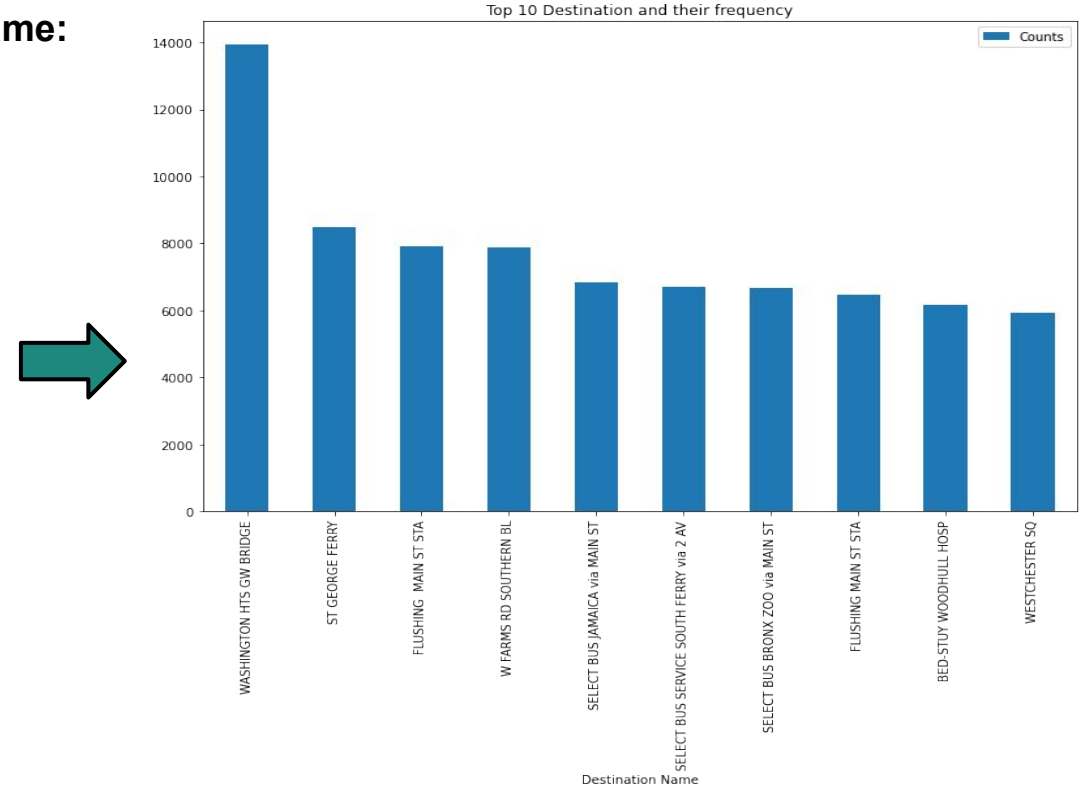


Exploratory Data Analysis (Cont.):

8. **Analysis variable 'Destination Stop':** Total number of Unique **destination stop** is **651.**

9. **Top 10 frequent destination stop name:**

	Destination Name	Counts
0	WASHINGTON HTS GW BRIDGE	13936
1	ST GEORGE FERRY	8480
2	FLUSHING MAIN ST STA	7903
3	W FARMS RD SOUTHERN BL	7891
4	SELECT BUS JAMAICA via MAIN ST	6846
5	SELECT BUS SERVICE SOUTH FERRY via 2 AV	6689
6	SELECT BUS BRONX ZOO via MAIN ST	6673
7	FLUSHING MAIN ST STA	6453
8	BED-STUY WOODHULL HOSP	6169
9	WESTCHESTER SQ	5942

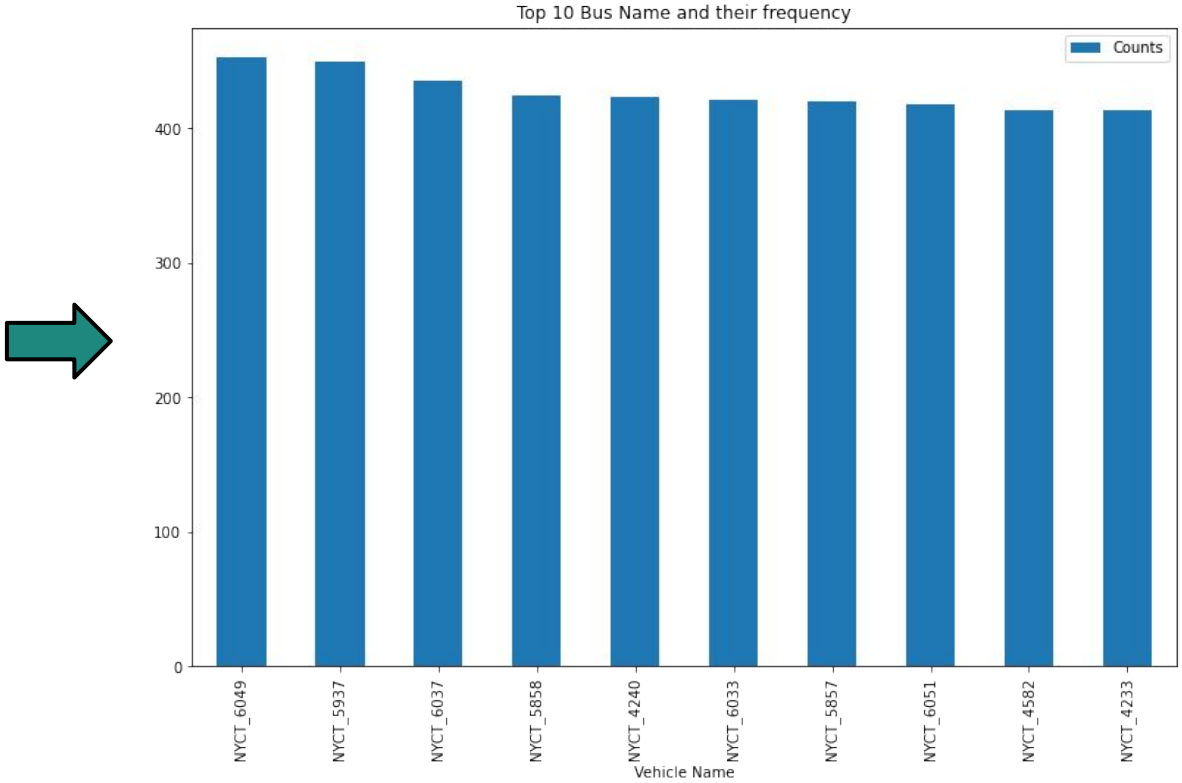


Exploratory Data Analysis (Cont.):

10. **Analysis variable 'Bus Name':** Total number of Unique **bus** is **4,629.**

11. **Top 10 frequent bus name:**

	Vehicle Name	Counts
0	NYCT_6049	452
1	NYCT_5937	449
2	NYCT_6037	435
3	NYCT_5858	424
4	NYCT_4240	423
5	NYCT_6033	421
6	NYCT_5857	420
7	NYCT_6051	417
8	NYCT_4582	413
9	NYCT_4233	413



Exploratory Data Analysis (Cont.):

12. **Analysis variable 'Arrival Approximation':** Total Unique **Arrival App.** is **208.**

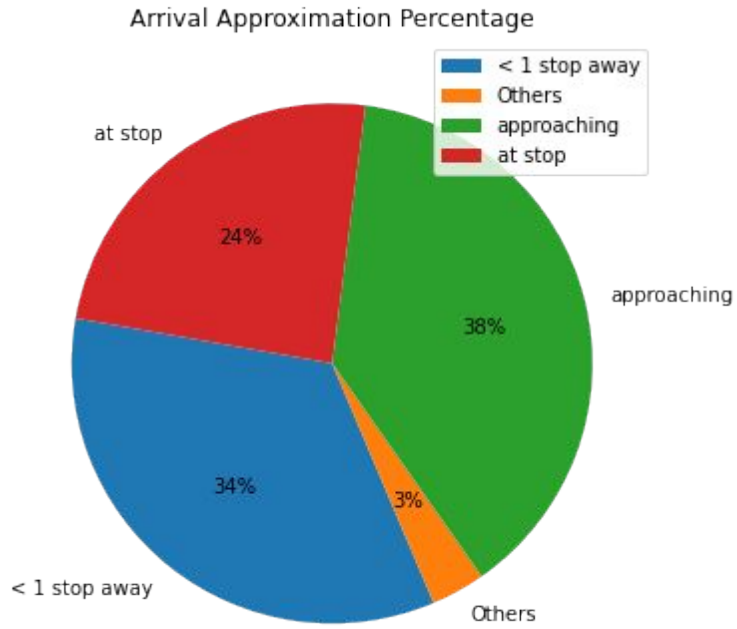
13. **Top 10 Arrival Approximation:**

	Arrival Approximation	Counts	Percentage
0	approaching	382261	38.2%
1	< 1 stop away	341566	34.2%
2	at stop	242694	24.3%
3	0.6 miles away	6034	0.6%
4	0.5 miles away	4493	0.4%
5	0.7 miles away	3805	0.4%
6	0.8 miles away	2127	0.2%
7	0.9 miles away	1723	0.2%
8	1.0 miles away	1489	0.1%
9	1.1 miles away	978	0.1%

Exploratory Data Analysis (Cont.):

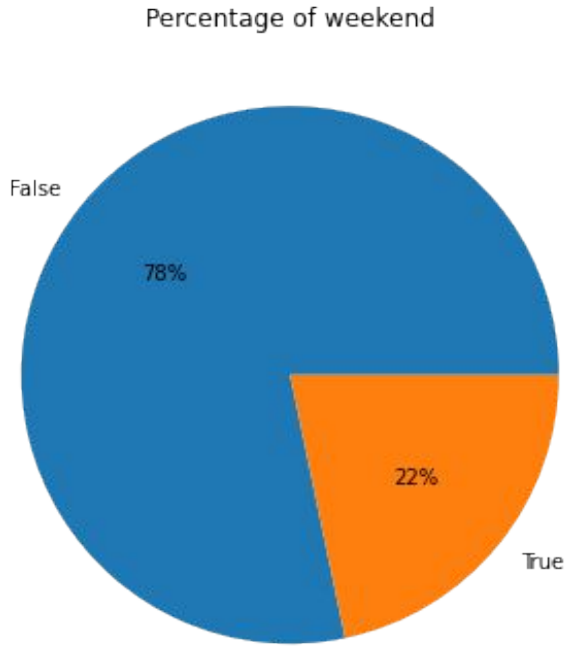
14. Visualize Arrival Approximation:

Arrival Approximation	Counts	Percentage
approaching	382261	38.2%
< 1 stop away	341566	34.2%
at stop	242694	24.3%
Others	33479	3.3%



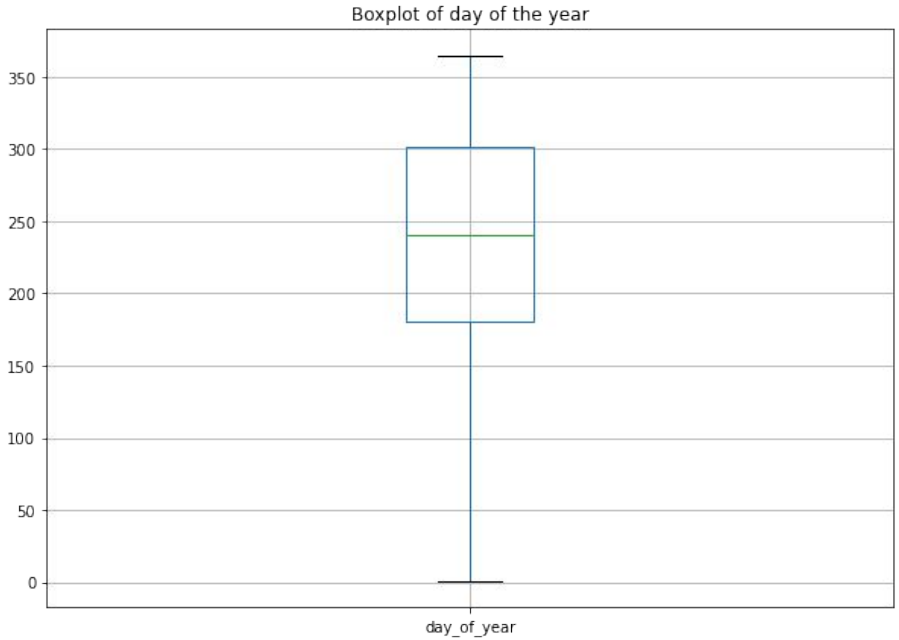
Exploratory Data Analysis (Cont.):

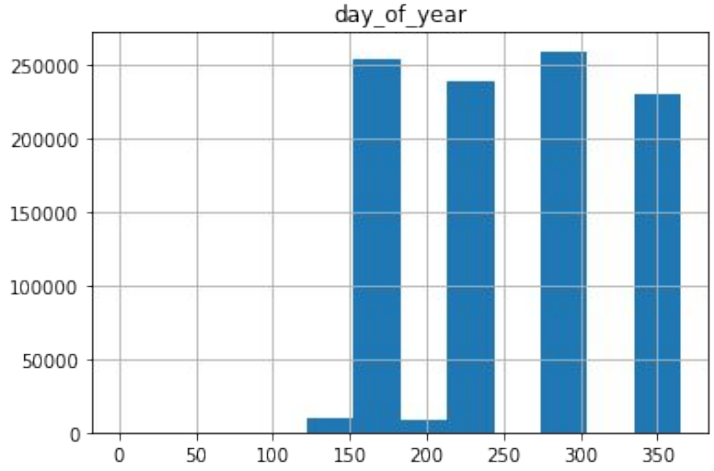
15. **Analysis variable ‘Weekend Status’:** Weekend status



Exploratory Data Analysis (Cont.):

16. Analysis variable ‘Day of Year’:






```
The volumns of month 6 is: 263491
The volumns of month 12 is: 229950
The volumns of month 10 is: 259155
The volumns of month 8 is: 247393
The volumns of month 11 is: 9
The volumns of month 1 is: 1
The volumns of month 7 is: 1
```

Exploratory Data Analysis (Cont.):

17. Most Important Feature of the project:

- line_name
- org_name
- dest_name
- vech_name
- day_of_year
- month_number
- day_of_month
- time_of_day
- weekend_status
- delays

Exploratory Data Analysis (Cont.):

18. Most Important Feature of the project:

	line_name	org_name	dest_name	vech_name	weekend_status	day_of_year	month_number	day_of_month	time_of_day	delays
0	B46-SBS	UTICA AV/AV N	SELECT BUS De KALB AV via UTICA	NYCT_7330	False	156	6	5	439	1.85
1	M101	LEXINGTON AV/E 100 ST	LTD EAST VILLAGE 6 ST via LEX AV	NYCT_6057	False	342	12	8	721	10.03
2	Bx31	TREMONT AV/LANE AV	WOODLAWN KATONAH AV	NYCT_7703	False	298	10	25	341	0.10
3	Q20A	COLLEGE PT BL/15 AV	JAMAICA MERRICK BL via 20 AV via MAIN S	NYCT_7391	False	278	10	5	825	0.15
4	Bx34	VALENTINE AV/E FORDHAM RD	WOODLAWN KATONAH AV	NYCT_4102	False	347	12	13	439	0.23

Implementation of ML Algorithm:

1. As the goal is to predict delay time (continuous value), I use regression.
2. For that, perform factorization to categorical variables.

	line_number	org_number	dest_number	vech_number	is_weekend	day_of_year	month_number	day_of_month	time_of_day	delays
0	0	0	0	0	0	156	6	5	439	1.85
1	1	1	1	1	0	342	12	8	721	10.03
2	2	2	2	2	0	298	10	25	341	0.10
3	3	3	3	3	0	278	10	5	825	0.15
4	4	4	2	4	0	347	12	13	439	0.23

3. **Train and Test Split :** Split the dataset 80-20 for train and test.

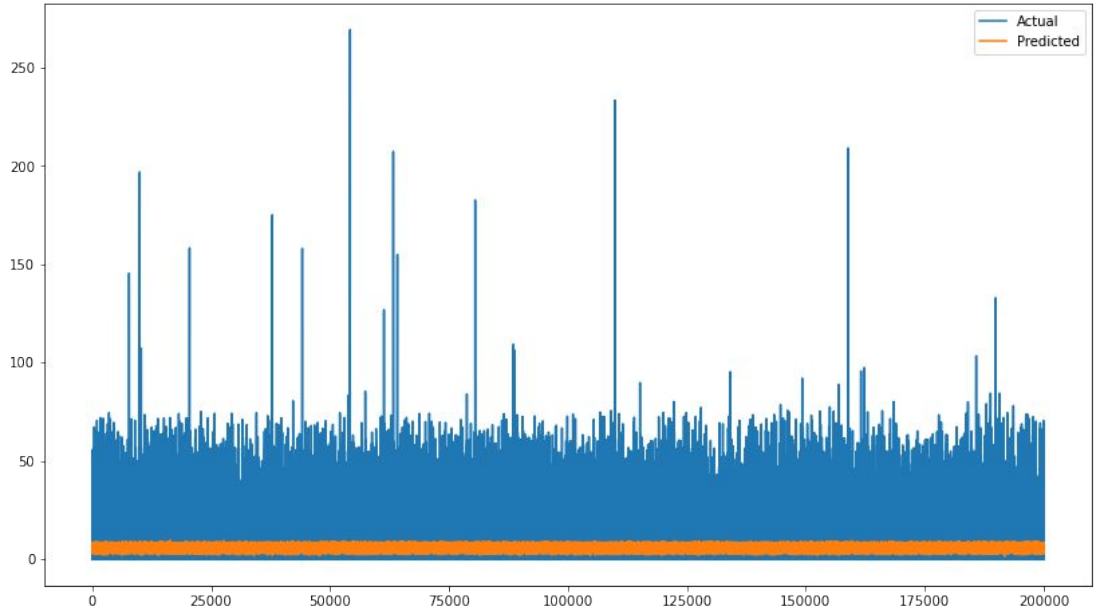
```
X Train (800000, 9), X Test: (200000, 9), Y train: (800000,), Y Test: (200000,)
```

Implementation of ML Algorithm:

4. Implement Linear Regression:

```
Coefficients: [-4.95098058e-03 -2.40001629e-03  8.87056957e-04  4.83572921e-05
  3.98040717e-01 -3.26490386e+00  9.95879609e+01  3.24205572e+00
  3.29442801e-03]

Mean squared error: 8.894963577608513
Coefficient of determination: 0.017925941050149108
```



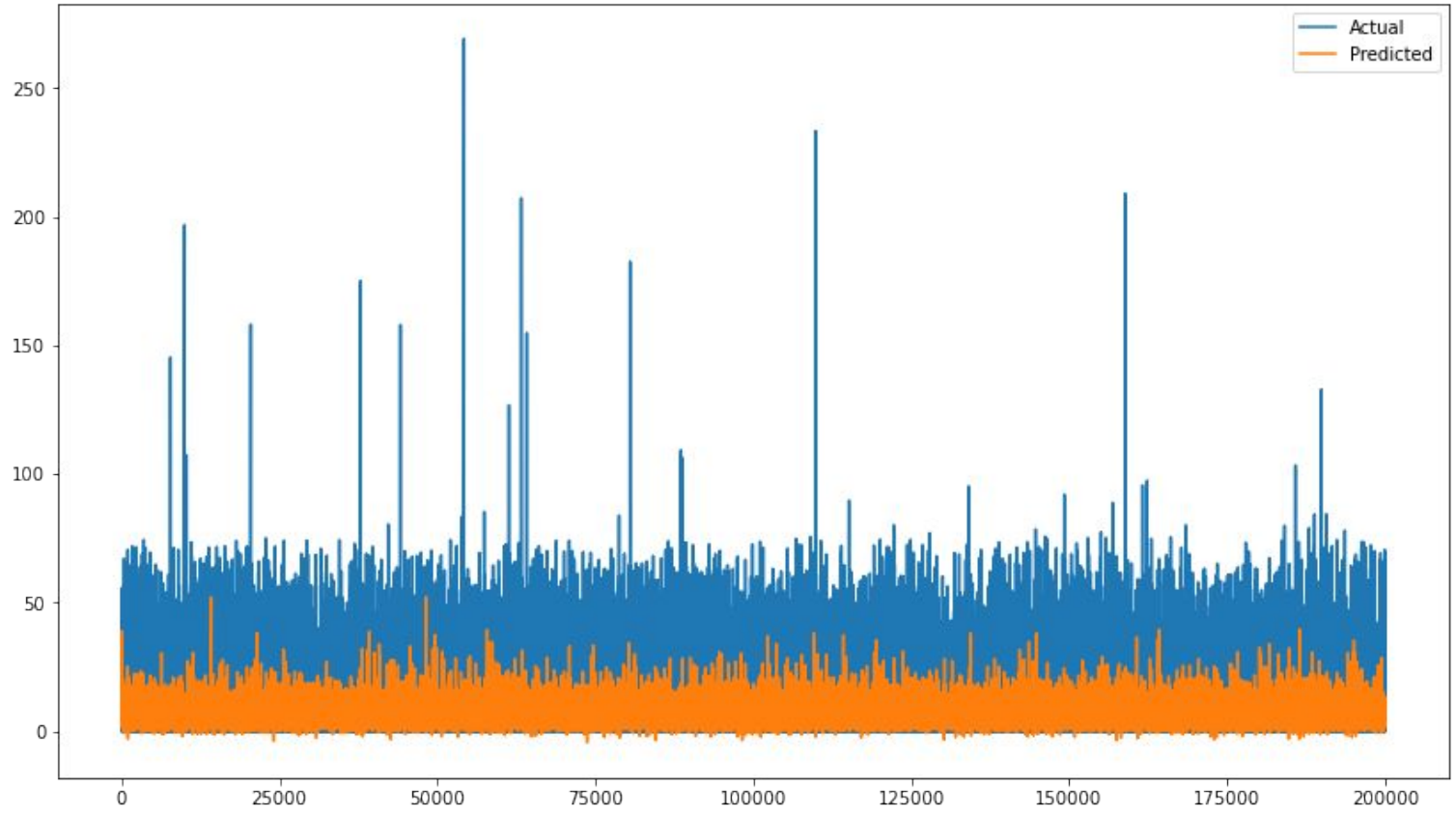
Implementation of ML Algorithm:

5. Implement XG-Boost Regression:

- Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.

- Final RMSE become: **8.35**

Implementation of ML Algorithm:



Implementation of ML Algorithm:

5. Hyperparameter Optimization of XG-Boost Regression:

- Use RandomizedSearchCV for hyperparameter tuning.

```
# Hyper parameter optimizations

n_estimators = [100, 500, 900, 1200]
max_depth = [2, 3, 5, 10]
booster = ['gbtree', 'gblinear']
learning_rate = [0.05, 0.1, 0.15, 0.20]
min_child_weight = [1, 2, 4]
base_score = [0.25, 0.5, 0.75, 1]

heyperparameter_grid = {
    'n_estimators': n_estimators,
    'max_depth': max_depth,
    'learning_rate': learning_rate,
    'min_child_weight':min_child_weight,
    'booster': booster,
    'base_score':base_score
}
```

```
# Initialize randomize search
random_cv = RandomizedSearchCV(estimator=clf_xgb,
                               param_distributions=heyperparameter_grid,
                               cv = 5,
                               n_iter=50,
                               scoring='neg_mean_absolute_error',
                               n_jobs=4,
                               verbose=5,
                               return_train_score=True,
                               random_state=42)
```

Implementation of ML Algorithm:

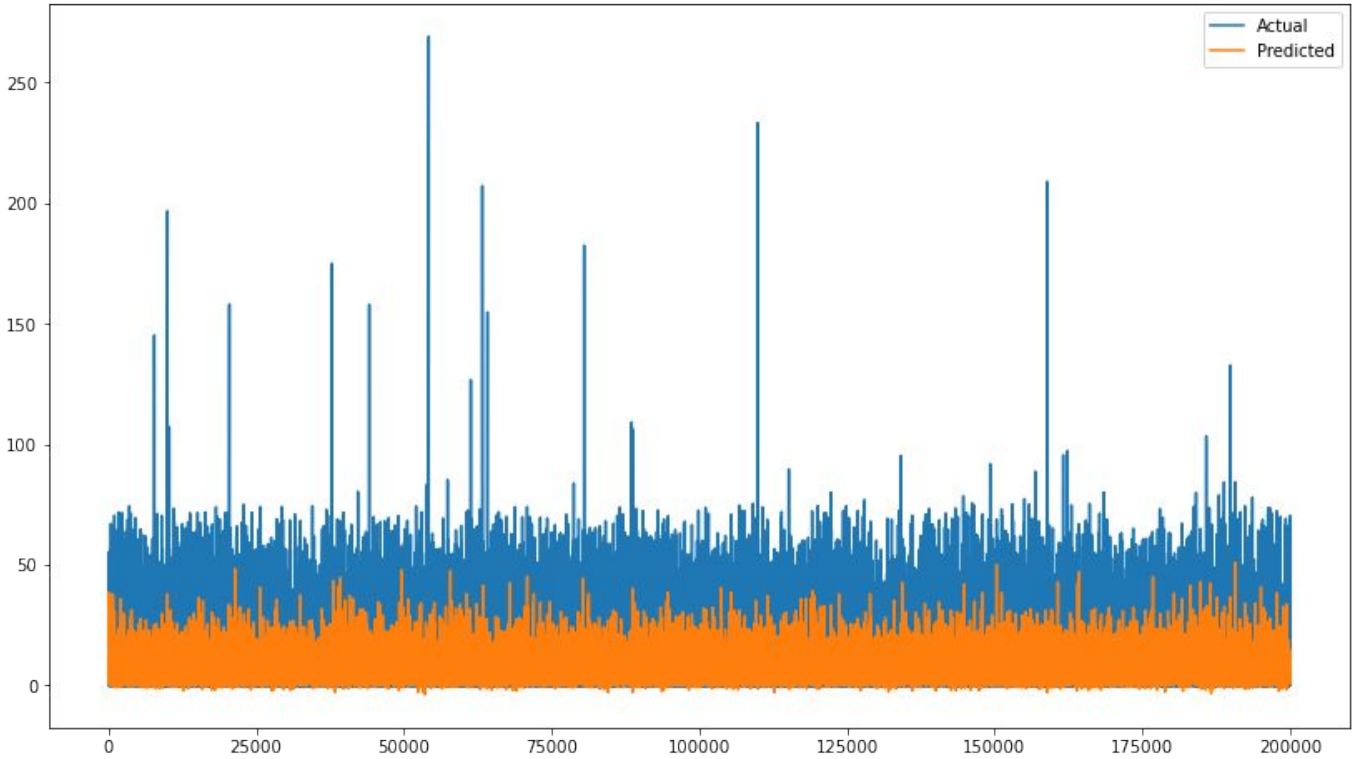
5. **Hyperparameter Optimization of** XG-Boost Regression:

- Best hyperparameter:

```
The Best parameters are: XGBRegressor(base_score=1, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, feature_types=None, gamma=0, gpu_id=-1,
             grow_policy='depthwise', importance_type=None,
             interaction_constraints='', learning_rate=0.05, max_bin=256,
             max_cat_threshold=64, max_cat_to_onehot=4, max_delta_step=0,
             max_depth=10, max_leaves=0, min_child_weight=4, missing=nan,
             monotone_constraints='()', n_estimators=900, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, ...)
```

- Final RMSE become: **8.09**

Implementation of ML Algorithm:



Result Analysis:

Approach	RMSE
Linear Regression	8.90
XG-Boost Regression	8.35
Tuned XG-Boost Regression	8.09

Code Repository:

Github: https://github.com/AhmedDiderRahat/ut_wise2223

Resources

- https://machinelearningmastery.com/xgboost-for-regression/
- https://www.kaggle.com/datasets/stoney71/new-york-city-transport-statistics?select=mta_1708.csv
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

Thank You